1. Activité : Mémento Docker

- Rechercher une image dans Docker Hub

```
docker search mot-clé
```

- Récupération d'une image dans sa dernière version

```
docker pull debian:latest
```

- Afficher les images Docker présentes sur la machine

```
docker images
docker image ls
```

- Lancement sans option d'un conteneur

```
docker run ubuntu
```

- Lister les conteneurs actifs

```
docker ps
```

- Lister tous les conteneurs, y compris les inactifs

```
docker ps -a
```

- Supprimer un conteneur arrêté

```
docker rm nom_du_conteneur
```

- Lancer un conteneur à partir d'une image avec une commande donnée

```
docker run [image à instancier] [commande à exécuter]
```

- Lancer un conteneur à partir d'une image en mode interactif -i et en attachant un shell -t

```
docker run -i -t ubuntu
```

- Lancer un conteneur à partir d'une image en mode interactif -i et en attachant un shell -t particulier

```
docker run -i -t debian /bin/bash
```

- Quitter un conteneur avec exit

```
conteneur# exit
```

- Trouver la liste des modifications d'un conteneur par rapport à son image de lancement

```
docker diff [identifiant du conteneur]
```

- Persister l'état d'un conteneur en une nouvelle image

```
docker commit [identifiant ou nom du conteneur] [nom de l'image]
```

- Supprimer une image (après avoir supprimer les conteneurs qui l'utilise) :

```
docker rmi [nom de l'image]
```

- Supprimer plusieurs images

```
docker rmi [image 1] [image 2] ... [image n]
```

- Lancer et supprimer un conteneur dès la sortie du processus (conteneurs à la durée de vie limitée à une exécution

```
docker run --rm hello-world
```

- Lancer un conteneur avec un nom donné

```
docker run --name=[nom fourni] [image à instancier]
```

- Modifier le hostname au démarrage d'un conteneur

```
docker run -h [hostname à utiliser] [image à instancier]
```

- Exemple

```
docker run -it –rm -h dhcpserv debian
```

- Lancer un conteneur en mode détaché

```
docker run -d [image à instancier]
```

- Remplacer une portion d'arborescence du conteneur par un emplacement hôte

```
docker run -v [dossier hôte]:[dossier sur conteneur] ...
```

- Injecter une variable d'environnement au démarrage d'un conteneur

```
docker run -e=[variable d'environnement] [image à instancier]
```

- Créer une variable d'environnement au démarrage d'un conteneur

```
docker run -e=[variable d'environnement=valeur] [image à instancier]
docker run -e [variable d'environnement=valeur] [image à instancier]
```

- Créer des variables d'environnement définies dans un fichier au démarrage d'un conteneur

```
docker run –env-file=[nomFichierContenantVariableEnvironnement&valeur] [image à instancier]
```

- Démarrer un conteneur arrêté en utilisant son nom, son nom généré ou son identifiant, même partiel :

```
docker start [conteneurID/NAME]
```

- Arrêter un conteneur en utilisant son nom, son nom généré ou son identifiant, même partiel :

```
docker stop [conteneurID/NAME]
```

- Redémarrer un conteneur en cours

```
docker restart [conteneurID/NAME]
```

- Recycler un conteneur en spécifiant la période de grâce

```
docker restart -t 20 [conteneur]
```

- Envoyer un signal d'arrêt à un conteneur

```
docker kill -s [code du signal] [conteneurID/NAME]
```

Par défaut, le code du signal est KILL, ce qui correspond à un arrêt du processus ciblé.

- Accédez à un conteneur démarré :

```
docker attach [containerID/NAME]
```

- Quitter un conteneur mais le laisser s'exécuter en arrière-plan

```
Ctrl + p + q.
```

- Supprimer un conteneur

```
docker rm [conteneurID/NAME]
```

- Supprimer un conteneur de manière forcée

```
docker rm -f [conteneurID/NAME]
```

- Supprimer tous les conteneurs

```
docker rm `docker ps -a -q` -> ne fonctionne pas sous Windows est-ce le ` ?
```

- La commande system et sa sous-commande prune permettant de réaliser le ménage dans les conteneurs arrêtés, les images orphelines et d'autres ressources a priori non utilisées. Ne supprime pas les images.

Faire le ménage dans les conteneurs arrêtés, images orphelines, et autres objets (network par exemple)

```
docker system prune
```

- Visualiser les logs d'un conteneur

```
docker logs [conteneur]
```

- Explorer les processus d'un conteneur

```
docker top [conteneur]
```

- Suivre l'évolution des ressources utilisées par un conteneur

```
docker stats [conteneur]
```

From:
https://siocours.lycees.nouvelle-aquitaine.pro/ - **Les cours du BTS SIO**

Permanent link:
https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/reseau/docker/memento

Last update: **2022/03/31 20:45**